Lucent Technologies
Bell Labs Innovations

Lucent VPN Firewall Bricks® 350, 1000

FIPS 140-2
Non-Proprietary Security Policy

Level 2 Validation
Version 6.0

August 30, 2004

Non-Proprietary Security Policy

Table of Contents

1 Introduction

1.1 Purpose

This is the non-proprietary Cryptographic Module Security Policy for the Brick 350 and Brick 1000. This security policy describes how the Brick 350 and Brick 1000 (Hardware Version: Brick 350 and Brick 1000; Firmware Version: Lucent LVF 7.1.189) meet the security requirements of FIPS 140-2, and how to operate the Bricks in a secure FIPS 140-2 mode. This policy was prepared as part of the Level 2 FIPS 140-2 validation of the Brick 350 and Brick 1000 VPN Firewalls.

FIPS 140-2 (Federal Information Processing Standards Publication 140-2 — *Security Requirements for Cryptographic Modules*) details the U.S. Government requirements for cryptographic modules. More information about the FIPS 140-2 standard and validation program is available on the NIST website at http://csrc.nist.gov/cryptval/.

1.2 References

This document deals only with operations and capabilities of the Brick 350 and Brick 1000 in the technical terms of a FIPS 140-2 cryptographic module security policy. More information is available on the Brick 350 and Brick 1000 and the entire Brick series, from the following sources:

- The Lucent Technologies website contains information on the full line of products at http://www.lucent.com. The Lucent product descriptions can be found at: http://www.lucent.com/products/subcategory/0,,CTID+2017-STID+10080-LOCL+1,00.html

- For answers to technical or sales related questions please refer to the contacts listed on the Lucent Technologies website at http://www.lucent.com/support/access.html.

- The NIST Validated Modules website (http://csrc.nist.gov/cryptval) contains contact information for answers to technical or sales-related questions for the module

1.3 Terminology

In this document, the Brick 350 and Brick 1000 as a group are referred to as the Module(s) or module(s). When referring to a specific Brick, the module is referred to as the Brick 350 module, or the Brick 1000 module.

1.4 Document Organization

The Security Policy document is part of the FIPS 140-2 Submission Package. In addition to this document, the Submission Package contains:

- Vendor Evidence document
- Finite State Machine
- Module Software Listing

- Other supporting documentation as additional references

This document provides an overview of the Brick 350 and Brick 1000 modules and explains the secure configuration and operation of the modules. This introduction section is followed by Section 2, which details the general features and functionality of the Brick 350 and Brick 1000 modules. Section 3 specifically addresses the required configuration for the FIPS-mode of operation.

This Security Policy and other Validation Submission Documentation was produced by Corsec Security, Inc. under contract to Lucent Technologies, Inc. With the exception of this Non-Proprietary Security Policy, the FIPS 140-2 Validation Submission Documentation is Lucent-proprietary and is releasable only under appropriate non-disclosure agreements. For access to these documents, please contact Lucent Technologies, Inc.

2 The Brick 350 and Brick 1000 VPN Firewalls

The VPN Firewall Brick is a high-speed packet-processing appliance, oriented towards providing security functions. The module is offered in several models, providing different physical interface combinations as well as different capacity and throughput ratings. The module is Intel Pentium based, using a PCI bus backplane, so its speed and capacity scales with standard components and has a minimum growth predictable according to Moore's Law. The Brick product line provides Local Area Network (LAN)-level Ethernet interfaces, in both 10/100 copper and Gigabit fiber ports. In the larger module (Brick 1000), the fan is the only continuously moving part, allowing for the module to have an extremely long hardware mean time between failures (MTBF) – greater than 7 years.

Within the module, local policy and configuration data are only stored on a solid-state Non-Volatile Random Access Memory (NVRAM) disk. The module does not run as an application on top of a commercial operating system; rather, it runs as the kernel of a small, highly application-specific operating system, designed for small embedded security applications.

VPN Firewall Bricks incorporate these features:

- Packet Forwarding – Bridging and Routing
- IEEE 802.1q VLAN Tag Support
- Virtual Firewalls & Stateful Packet Filtering
- Application Filters
- Virtual Private Networking (VPN) & Network Address Translation (NAT)
- User Authentication
- Quality of Service/Bandwidth Management
- Denial of Service Protection
- Brick Partitions
- Brick Failover/Redundancy & State Sharing
- Dynamic Address Support
- Logging

The same software binary image ("tvpc.Z") runs on all modules, so all features discussed are available on all module platforms. The binary images are identical across all platforms, regardless of the Brick's model number or configuration setup.

Bricks are available in a variety of hardware models; the models differ solely in throughput, capacity, and physical interface types. This Security Policy applies to the following FIPS 140-2 Level 2 validated Modules:

Brick 350 Module: For enterprise-class demands of large corporate facilities.

- VPN Firewall Brick® Model 350 Basic [8-10/100 Ethernet Ports, Internal AC Power Supply, Internal Floppy Drive]

Brick 1000 Module: For service providers offering advanced security services packages.

- VPN Firewall Brick® Model 1000 (5/4) [5-10/100 Ethernet Ports/4-Gigabit Fiber Ports, Dual Internal AC Power Supply, Internal Floppy Drive]

2.1 The Cryptographic Module

Figure 1 - The Brick 350 Module

Figure 2 - The Brick 1000 Module

The Brick 350 and Brick 1000 modules are multiple-chip standalone cryptographic modules. The cryptographic boundary is defined as the front, right, left, top, and bottom sides of the case; all portions of the rear of the case that are not designed to accommodate a network module or power supply; and the inverse of the three-dimensional space within the case that would be occupied by any installed power supply or network module that does not perform approved services. The cryptographic boundary includes the connection apparatus between the network modules and power supplies and the motherboard that hosts the network modules and power supplies, but the boundary does not include the power supplies and network modules themselves. In other words, the cryptographic boundary encompasses all hardware components within the case of the module except any installed network modules and power supplies. All of the functionality discussed in this document is provided by components within this cryptographic boundary.

The Brick 1000 module requires that a special opacity shield be installed on the top portion of the rear of the module, covering the top row of ventilation holes along the rear of the chassis (as shown in Figure 3) in order to operate in FIPS-approved mode. The shield completely covers the ventilation holes on the top of the rear panel of the Brick 1000 module. To apply, remove the three pan-head screws from the rear of the chassis and attach the opacity shield to the chassis, using the three flat-head screws that are supplied with the FIPS kit. Figure 3 demonstrates the proper application of the shield.

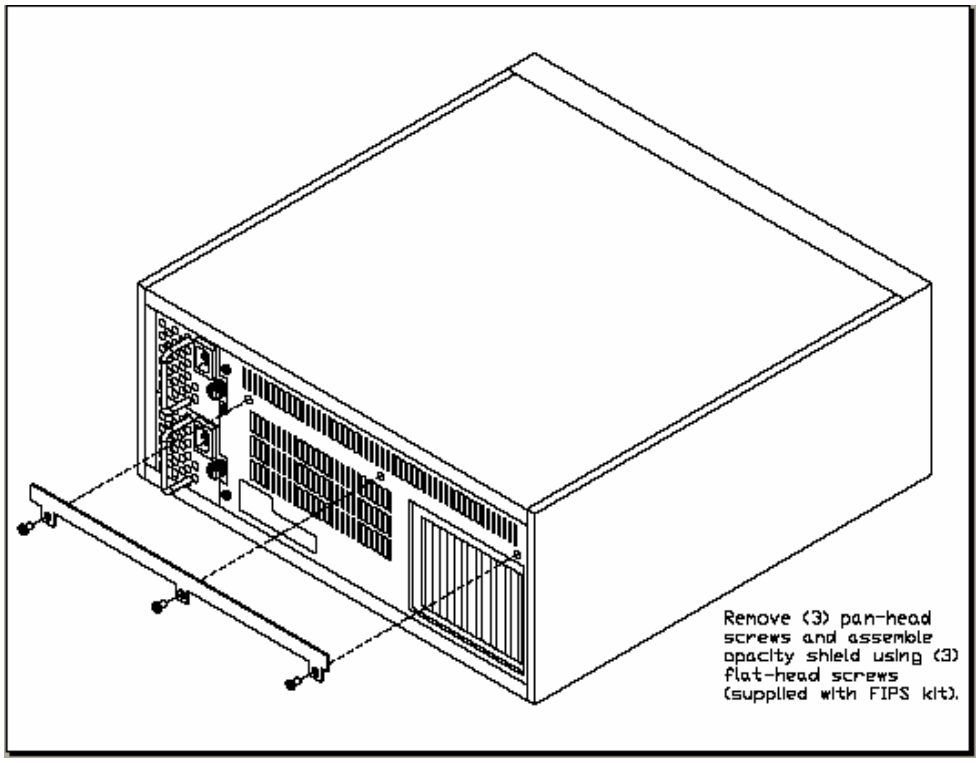

Figure 3 – Brick 1000 Opacity Shield Application

2.2 Module Interfaces

Module features such as tunneling, data encryption, and termination of Remote Access Wide Area Networks (WANs) via Internet Protocol Security (IPSec) make the Lucent VPN Firewall Brick an ideal platform for building virtual private networks. The interfaces for the module are located on the front and rear panels of the modules as shown in the following figures.

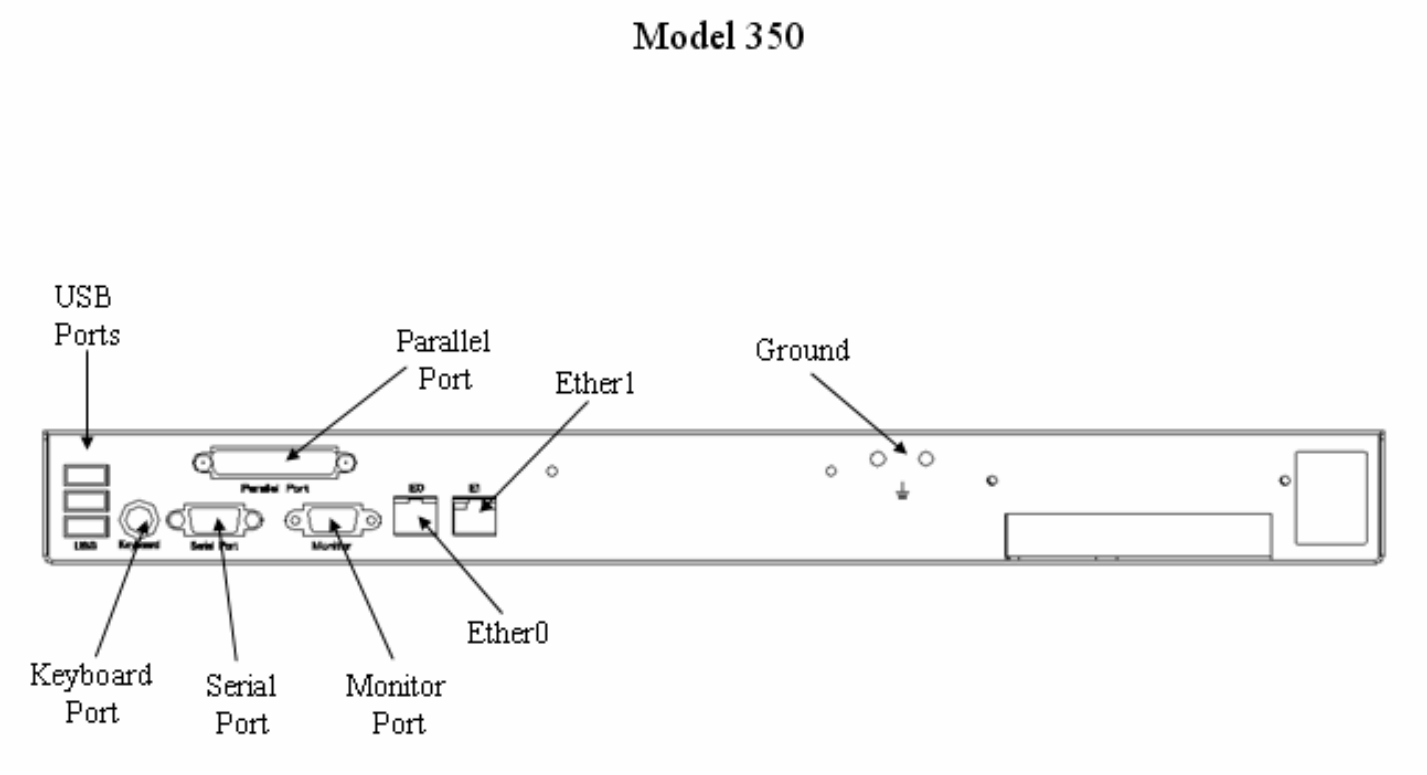

Figure 4 – Brick 350 Physical Interfaces

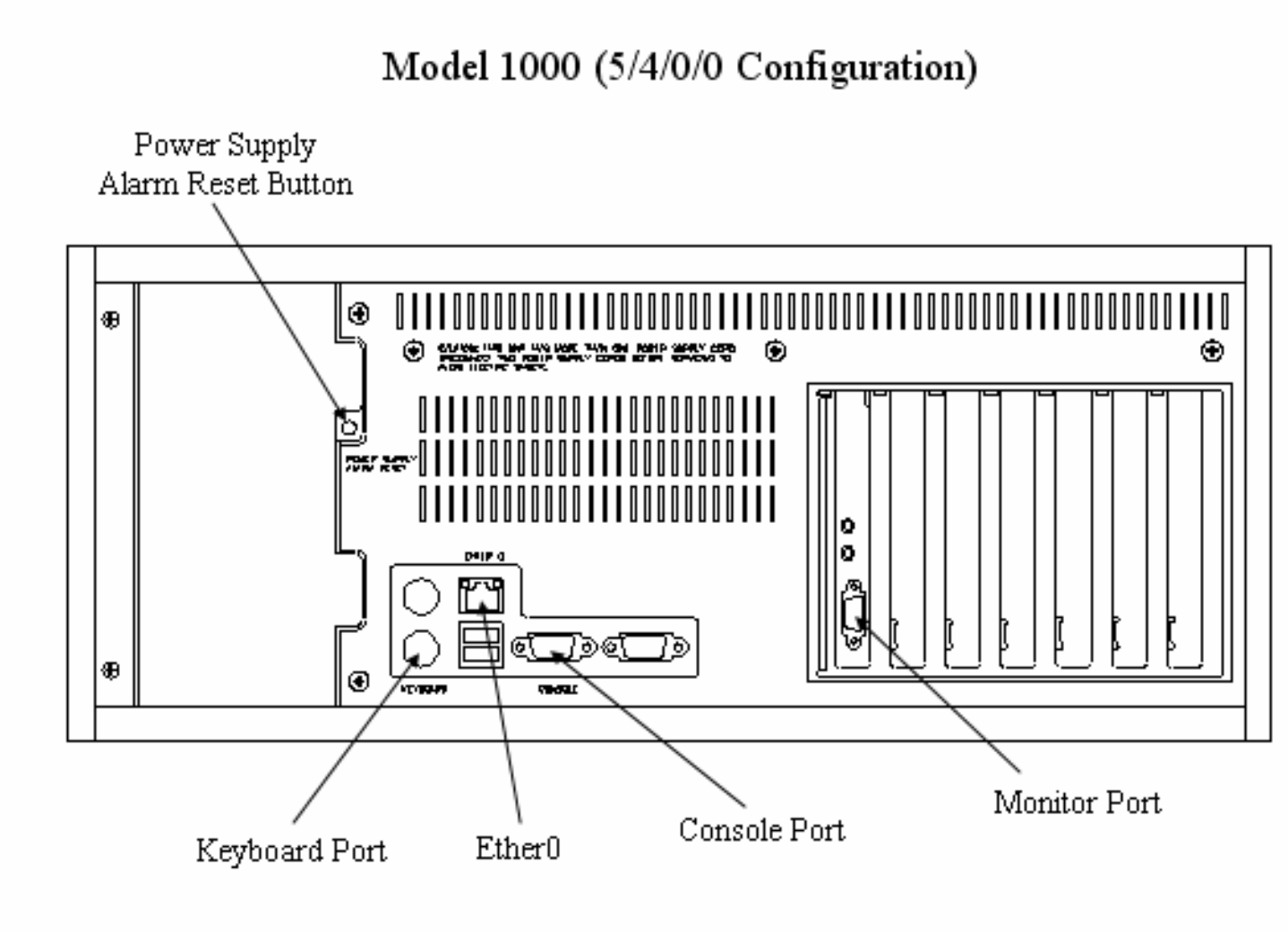

Figure 5 – Brick 1000 Physical Interfaces

The physical interfaces include a power switch, a keyboard port, a monitor port, and a console port (RS-232 serial connector) on the backplane for local system access (on the Brick 350, the port labeled “Serial Port” is the Console Port), Ethernet ports (Ether0 and Ether1 for the Brick 350, and Ether0 for the Brick 1000), and the Network Module connection interfaces on the motherboard.

The module’s status interfaces are located on the front panel. These LEDs provide overall status of the module’s operation. Figure 6 and Figure 7 show the front panel LEDs of the Brick 350 and Brick 1000 modules. Table 1 and Table 2 provide descriptions for the front panel LEDs, Table 3 and Table 4 provide descriptions for the rear panel LEDs, and Table 5 provides a description of the modules’ audible buzzer.

Front Panel LEDs:

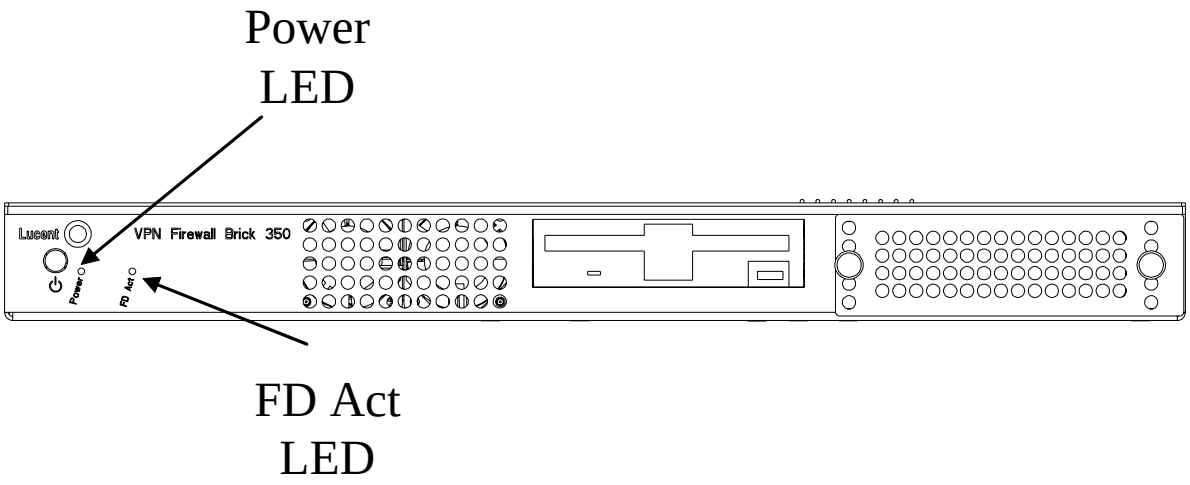

Figure 6 – Brick 350 Front Panel LEDs

Model 1000 – Front View (Cover Open)

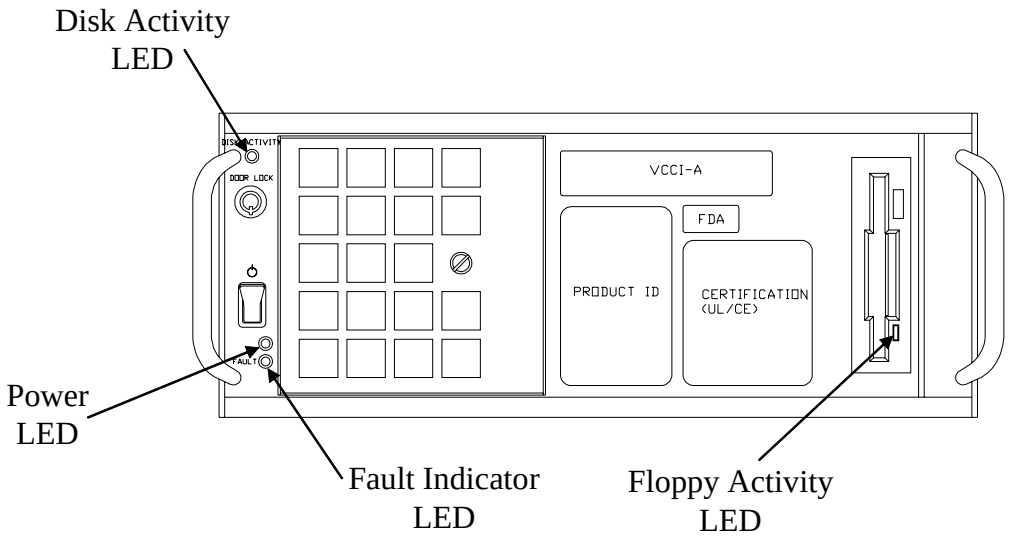

Figure 7 – Brick 1000 Front Panel LEDs

LED	Indication	Description
Power	Solid	Power is supplied to the module
	Off	The module is not powered on
FD Act	Intermittent	The flash disk is in use
	Off	The flash disk is not in use
Floppy Drive	On	The floppy drive is reading a diskette
	Off	The floppy drive is not in use

Table 1 – Brick 350 Front Panel LEDs and Descriptions

LED	Indicator	Description
Power	Green	Power is supplied to the module
	Off	The module is not powered on
Floppy Drive	On	The floppy drive is reading a diskette
	Off	The floppy drive is not in use
Disk Activity	Amber	The flash disk is in use
	Off	The flash disk is not in use
Fault (Power Supply)	Orange	Power supply failure
	Off	The power supplies are on and functioning

Table 2 – Brick 1000 Front Panel LEDs and Descriptions

Rear Panel LEDs:

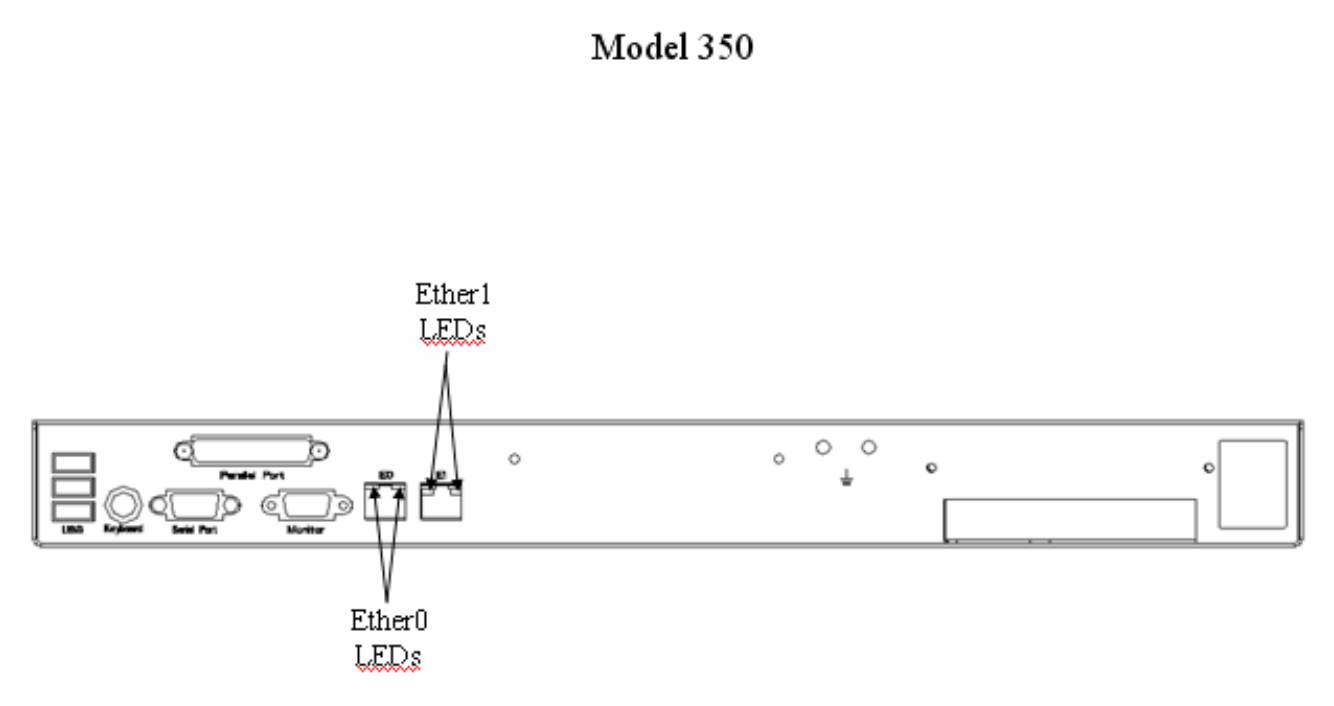

Figure 8 – Brick 350 Rear Panel LEDs

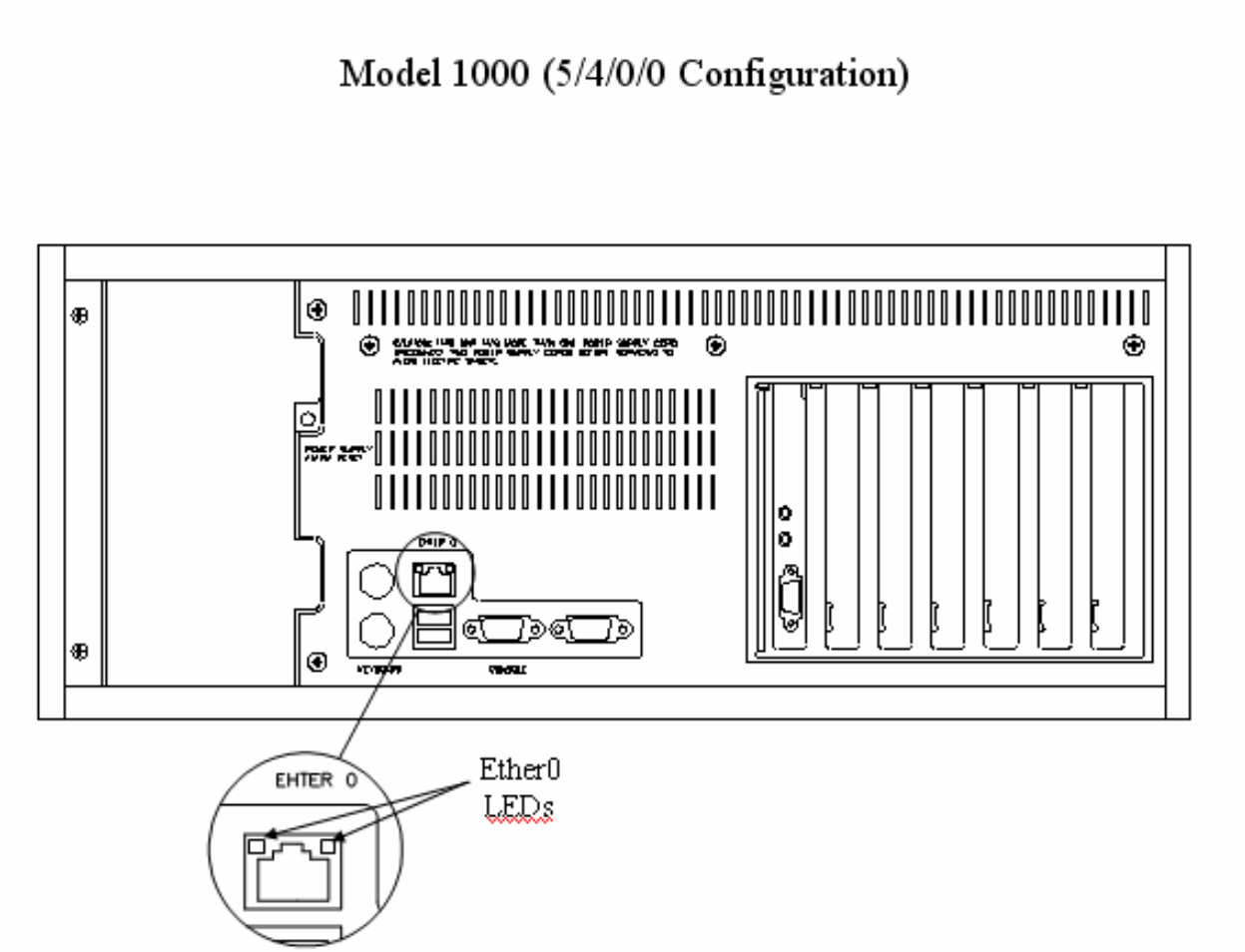

Figure 9 – Brick 1000 Rear Panel LEDs

LED	Indicator	Description
Motherboard E0	Left: Off	Port connected at 10Mbps
	Left: On	Port connected at 100Mbps
	Right: On	Port is on
	Right: Intermittent	Data being transferred
Motherboard E1	Left: Off	Port connected at 10Mbps
	Left: Green	Port connected at 100Mbps
	Left: Yellow	Port connected at 1000Mbps
	Right: On	Port is on
	Right: Intermittent	Data being transferred

Table 3 - Brick 350 Rear Panel LEDs and Descriptions

LED	Indicator	Description
Encryption Accelerator (LED)	Blinking	Encryption Accelerator Card is in use
	Solid	Encryption Accelerator Card failed while LED was blinking in the ON state
	Off	If Encryption Accelerator Card is installed, either the EAC is not currently in use or the EAC failed while LED was blinking in the OFF state
Motherboard Ethernet Port 0	Left: Off, Right: On	Good connection at 10Mbps
	Left: On, Right: On	Good connection at 100Mbps
	Left: Off, Right: Off	No connection
	Left: Off, Right: Intermittent	Data being transferred at 10Mbps
	Left: On, Right: Intermittent	Data being transferred at 100Mbps

Table 4 – Brick 1000 Rear Panel LEDs and Descriptions

Audible	Indicator	Description
Buzzer	Sustained alarm	A power supply has failed
	Beep	OS image has successfully been loaded by floppy
	Off	Alarm Cut Off Switch is enabled or the module is powered off

Table 5 - Brick 350 and Brick 1000 Module Audible Description

All of these physical interfaces are separated into the logical interfaces from FIPS 140-2 as described in the following tables:

Brick 1000 Module Physical Interface	Brick 1000 Module FIPS 140-2 Logical Interface
Network Module Interface Ethernet Port Console Port Floppy Drive PS/2 Keyboard Port	Data Input Interface
Network Module Interface Ethernet Port SVGA Video Port Console Port	Data Output Interface
Network Module Interface Ethernet Port Power Switch Power Supply Alarm Reset Button PS/2 Keyboard Port Console Port	Control Input Interface
Network Module Interface Ethernet Port	Status Output Interface

Brick 1000 Module Physical Interface	Brick 1000 Module FIPS 140-2 Logical Interface
SVGA Video Port Ethernet Port LEDs Disk Activity LED Fault Status Indicator LED Power LED Floppy Drive LED Buzzer	
Motherboard	Power Interface
USB Port #1 USB Port #2 Serial Port Parallel Port Monitor Port #2 (Motherboard) Sound Ports Mouse Port	Disabled / Non-functional

Table 6 - Brick 1000 Module FIPS 140-2 Logical Interfaces

Brick 350 Module Physical Interface	Brick 350 Module FIPS 140-2 Logical Interface
Network Module Interface Ethernet Ports Serial Port Floppy drive PS/2 Keyboard Port	Data Input Interface
Network Module Interface Ethernet Ports SVGA Video Port	Data Output Interface
Network Module Interface Ethernet Ports Power Button PS/2 Keyboard Port	Control Input Interface
Network Module Interface Ethernet Ports SGVA Video Port Ethernet Port LEDs Flash Disk Activity LED Power LED Floppy Drive LED Buzzer	Status Output Interface
Motherboard	Power Interface
Parallel Port USB Port #1	Disabled / Non-functional

Brick 350 Module Physical Interface	Brick 350 Module FIPS 140-2 Logical Interface
USB Port #2 USB Port #3	

Table 7 - Brick 350 Module FIPS 140-2 Logical Interfaces

2.3 Roles and Services

Authentication is role-based. The two roles allowed in a FIPS 140-2 Level 2 approved mode of operation are the Crypto Officer role and the User role. The Crypto Officer (via the Lucent Security Management Server [LSMS]) generates a digital certificate which is then loaded into the module at initialization. This certificate is then used during a Secure Sockets Layer (SSL)-like protocol to authenticate the Crypto Officer to the module during all future authentication attempts. Users authenticate to the module using a shared secret Hashed Message Authentication Code - Secure Hash Algorithm (HMAC-SHA-1) key. This authentication is per packet via verification of an HMAC.

The Crypto Officer communicates with the module through an encrypted session that is established using the Crypto Officer Session Keys (DES or 3DES – NIST FIPS PUB 46-3 and HMAC – NIST PUB 198) and authenticates to the module using a digital certificate. Virtual Private Network (VPN) functionality is available via the User Role. VPN clients authenticate to the module per (network-layer) packet using a shared secret HMAC-SHA-1 key configured by the Crypto Officer.

The Crypto Officer may also authenticate to the cryptographic module via the local console port using a password (which is hashed locally) in order to perform a small number of maintenance activities.

2.3.1 Crypto Officer Services

The Crypto Officer is responsible for the configuration and management of the module. The Crypto Officer first provides an initial configuration for the module and then is able to access the module over an encrypted session. Through this session, the Crypto Officer can perform full management of the module, including loading IPSec Security Associations (SAs) onto the module for Users.

During the initial configuration of the module, the Crypto Officer generates a disk using the LSMS and this information is then loaded onto the module over the Module's floppy disk drive. The files on this disk include the following configuration information:

- Crypto Officer certificate containing the Crypto Officer Certificate Authority (CA) Digital Signature Algorithm (DSA) public key
- DSA key pair for the module (the public key is contained in a certificate generated by the Crypto Officer)
- Diffie-Hellman (DH) public parameters
- IP address of the LSMS

- Domain Name Server (DNS) Host Name given to identify the Module

The module's public key (of the DSA key pair loaded onto the module) is contained in a certificate generated by the LSMS CA. Each module is given such a unique certificate, and this is used during the Crypto Officer handshake protocol to authenticate the module to the Crypto Officer. Additionally, the Crypto Officer possesses a certificate, to allow the module to authenticate the Crypto Officer. Collectively, these certificates provide a mutual authentication between the Crypto Officer and every module, so an intruder cannot masquerade as either the Crypto Officer or a module.

Once the module has been initialized, the Crypto Officer may begin management of the module through a Triple Data Encryption Standard (3DES) encrypted IP session. The module provides the Crypto Officer role exclusively to the LSMS after the initial configuration is completed. Digital certificates are used to authenticate the Crypto Officer to the module and the module to the Crypto Officer, and a Diffie-Hellman key agreement is performed to negotiate encrypted session keys (HMAC SHA-1 and 3DES keys). After the encrypted session is established, the Crypto Officer accesses the module's services through this session.

Through an encrypted session, the Crypto Officer configures the module for use by IPSec clients. The Crypto Officer loads IPSec SAs onto the module over the encrypted session, including any IPSec SA session keys. As part of these SAs, the Crypto Officer configuration shared secret HMAC keys used to authenticate the User to the module.

An operator assuming the Crypto Officer role performs all administrative functions listed below, which are services that are embedded within the LSMS and activated from Application Programming Interface (API) calls to the module:

Writing Commands…			
LSMS Function	**Service Call**	**Description**	**Service Output**
BTABLE	"begin tableload"	Prepare the module to download a full policy definition including both all of the individual rule policies and the brick configuration (routes, interfaces, VLANs, etc).	if the returned value is equal to the exact length of the issued command, then the command executed successfully; if the returned value is equal to any value other than the exact length of the issued command, then the command did not execute successfully.
BATABLE	"begin tableadd"	make a copy of the current brick zone table configuration in preparation for loading the initial (post-boot) policy for contacting the LSMS to download the initial policy. The reason for the copy is so that we do not lose state information in the event that we just transitioned from the standby to the active.	if the returned value is equal to the exact length of the issued command, then the command executed successfully; if the returned value is equal to any value other than the exact length of the issued command, then the command did not execute successfully.
[BLOAD]	"begin load"	Clears out any loading state from a zone in preparation for loading a new zone policy.	if the returned value is equal to the exact length of the issued command, then the

Writing Commands…			
LSMS Function	**Service Call**	**Description**	**Service Output**
			command executed successfully; if the returned value is equal to any value other than the exact length of the issued command, then the command did not execute successfully.
[STABLE]	"sign table"	saves full policy signer information (e.g. administrator name, date).	if the returned value is equal to the exact length of the issued command, then the command executed successfully; if the returned value is equal to any value other than the exact length of the issued command, then the command did not execute successfully.
[SDOMAIN]	"sign domain"	saves domain (zone) signer information (e.g. administrator name, date).	if the returned value is equal to the exact length of the issued command, then the command executed successfully; if the returned value is equal to any value other than the exact length of the issued command, then the command

Writing Commands…			
LSMS Function	**Service Call**	**Description**	**Service Output**
			did not execute successfully.
[ALOAD]	"abort load"	change brick state to "aborted" for use by the "read load state" command.	if the returned value is equal to the exact length of the issued command, then the command executed successfully; if the returned value is equal to any value other than the exact length of the issued command, then the command did not execute successfully.
[ETABLE]	"end tableload"	signals the end of a full load (prerequisite "begin tableload"). This causes the brick to verify the signatures on the load.	if the returned value is equal to the exact length of the issued command, then the command executed successfully; if the returned value is equal to any value other than the exact length of the issued command, then the command did not execute successfully.
[ELOAD]	"end load"	signals the end of a policy (prerequisite "begin load"). This causes the brick to verify the signatures on the policy.	if the returned value is equal to the exact length of the issued command, then the command

Writing Commands…			
LSMS Function	**Service Call**	**Description**	**Service Output**
			executed successfully; if the returned value is equal to any value other than the exact length of the issued command, then the command did not execute successfully.
[SWITCH]	"switch over"	make the pending full policy or individual zone policy active. (prerequisite begin load or begin tableload).	if the returned value is equal to the exact length of the issued command, then the command executed successfully; if the returned value is equal to any value other than the exact length of the issued command, then the command did not execute successfully.
[ATABLE]	"add table"	add an entry to the zone assignment table (prerequisite "begin tableload")	if the returned value is equal to the exact length of the issued command, then the command executed successfully; if the returned value is equal to any value other than the exact length of the issued command, then the command did not execute

Writing Commands…			
LSMS Function	**Service Call**	**Description**	**Service Output**
			successfully.
(none)	"adm cert"	passes the public certificate for the administrator signing this particular object. (prerequisite, begin load or tableload).	if the returned value is equal to the exact length of the issued command, then the command executed successfully; if the returned value is equal to any value other than the exact length of the issued command, then the command did not execute successfully.
(none)	"adm pk"	passes the signing administrators public key..(prerequisite, begin load or tableload).	if the returned value is equal to the exact length of the issued command, then the command executed successfully; if the returned value is equal to any value other than the exact length of the issued command, then the command did not execute successfully.
(none)	"data cert"	pass the public certificate (i.e. the signature) of the object (full load or individual zone load). (prerequisite, begin load or tableload).	if the returned value is equal to the exact length of the issued command, then the command executed

Writing Commands…			
LSMS Function	**Service Call**	**Description**	**Service Output**
			successfully; if the returned value is equal to any value other than the exact length of the issued command, then the command did not execute successfully.
[AETHTYP]	"add ethertype"	add an entry to the list of ethertype non-ip protocols allowed to pass through the firewall (prerequisite, begin tableload).	if the returned value is equal to the exact length of the issued command, then the command executed successfully; if the returned value is equal to any value other than the exact length of the issued command, then the command did not execute successfully.
[SETHTYP]	"switch ethertype"	active the pending list of ethertype non-ip protocols allowed to pass (prerequisite, begin tableload).	if the returned value is equal to the exact length of the issued command, then the command executed successfully; if the returned value is equal to any value other than the exact length of the issued command, then the command did not execute successfully.

Writing Commands…			
LSMS Function	**Service Call**	**Description**	**Service Output**
[ADSAP]	"add dsap"	add an entry to the list of dsap non-ip protocols allowed to pass through the firewall (prerequisite, begin tableload).	if the returned value is equal to the exact length of the issued command, then the command executed successfully; if the returned value is equal to any value other than the exact length of the issued command, then the command did not execute successfully.
[SDSAP]	"switch dsap"	activate the pending list of dsap non-ip protocols allowed to pass (prerequisite, begin tableload).	if the returned value is equal to the exact length of the issued command, then the command executed successfully; if the returned value is equal to any value other than the exact length of the issued command, then the command did not execute successfully.
[AROUTE]	"add route"	add an entry to the pending IP static routing table. (prerequisite, begin tableload).	if the returned value is equal to the exact length of the issued command, then the command executed successfully; if the

Writing Commands…			
LSMS Function	**Service Call**	**Description**	**Service Output**
			returned value is equal to any value other than the exact length of the issued command, then the command did not execute successfully.
[APROXY]	"add proxy"	add an entry to the pending reflection proxy table. (prerequisite, begin tableload).	if the returned value is equal to the exact length of the issued command, then the command executed successfully; if the returned value is equal to any value other than the exact length of the issued command, then the command did not execute successfully.
[ADPROXY]	"add dynamic proxy"	add an entry to the *active* reflection proxy table. (This is an old command that is no longer used in LVF version 7.1.189)	[This function cannot be used in the FIPS mode of operation.]
[DDPROXY]	"delete dynamic proxy"	delete an entry from the *active* reflection proxy table. (Never used.)	if the returned value is equal to the exact length of the issued command, then the command executed successfully; if the returned value is equal to any value other than the exact length of the

Writing Commands…			
LSMS Function	**Service Call**	**Description**	**Service Output**
			issued command, then the command did not execute successfully.
[ARULE]	"add rule"	adds a pending rule to the loading domain. (prerequisite, begin load).	if the returned value is equal to the exact length of the issued command, then the command executed successfully; if the returned value is equal to any value other than the exact length of the issued command, then the command did not execute successfully.
[ADRULE]	"add dynamic rule"	adds an active rule to the specified domain. (Never used.)	if the returned value is equal to the exact length of the issued command, then the command executed successfully; if the returned value is equal to any value other than the exact length of the issued command, then the command did not execute successfully.
[DDRULE]	"delete dynamic rule"	does nothing.	Does nothing
[AMASK]	"add mask"	adds a pending dependency mask to the specified	if the returned value is equal to

Writing Commands…			
LSMS Function	**Service Call**	**Description**	**Service Output**
		domain. (prerequisite, begin load).	the exact length of the issued command, then the command executed successfully; if the returned value is equal to any value other than the exact length of the issued command, then the command did not execute successfully.
[ADMASK]	"add dynamic mask"	adds an active dependency mask to the specified domain.	if the returned value is equal to the exact length of the issued command, then the command executed successfully; if the returned value is equal to any value other than the exact length of the issued command, then the command did not execute successfully.
[AHOST]	"add hostgrp"	adds a pending host group entry to the specified domain. (prerequisite, begin load)	if the returned value is equal to the exact length of the issued command, then the command executed successfully; if the returned value is equal to any value other than the

Writing Commands…			
LSMS Function	**Service Call**	**Description**	**Service Output**
			exact length of the issued command, then the command did not execute successfully.
[ADHOST]	"add dynamic hostgrp"	adds an active host group entry to the specified domain.	if the returned value is equal to the exact length of the issued command, then the command executed successfully; if the returned value is equal to any value other than the exact length of the issued command, then the command did not execute successfully.
[DDHOST]	"delete dynamic hostgrp"	deletes a host group entry from the specified domain. (Host group entry must have been loaded with an add dynamic hostgroup).	if the returned value is equal to the exact length of the issued command, then the command executed successfully; if the returned value is equal to any value other than the exact length of the issued command, then the command did not execute successfully.
[ASRV]	"add srvgrp"	adds a pending service group entry to the specified domain. (prerequisite,	if the returned value is equal to the exact length of

Writing Commands…			
LSMS Function	**Service Call**	**Description**	**Service Output**
		begin load).	the issued command, then the command executed successfully; if the returned value is equal to any value other than the exact length of the issued command, then the command did not execute successfully.
[ADSRV]	"add dynamic srvgrp"	adds an active service group entry to the specified domain. (Not used)	if the returned value is equal to the exact length of the issued command, then the command executed successfully; if the returned value is equal to any value other than the exact length of the issued command, then the command did not execute successfully.
[SCOMM]	"set comm"	sets file descriptor and address of the connection to the audit server.	if the returned value is equal to the exact length of the issued command, then the command executed successfully; if the returned value is equal to any value other than the exact length of the

Writing Commands…			
LSMS Function	**Service Call**	**Description**	**Service Output**
			issued command, then the command did not execute successfully.
[DISABLE]	"disable firewall"	turns off packet processing for packets not originating on the firewall or destined to the firewall.	if the returned value is equal to the exact length of the issued command, then the command executed successfully; if the returned value is equal to any value other than the exact length of the issued command, then the command did not execute successfully.
[RENABLE]	"reenable firewall	undoes "disable firewall". firewall or destined to the firewall.	if the returned value is equal to the exact length of the issued command, then the command executed successfully; if the returned value is equal to any value other than the exact length of the issued command, then the command did not execute successfully.
[RFRSHMAC]	"refresh mac table"	marks all of the MAC table entries as stale so that they can move if necessary. Any sessions that have a	if the returned value is equal to the exact length of the issued

Writing Commands…			
LSMS Function	**Service Call**	**Description**	**Service Output**
		pointer to this entry have to be rerouted the next time a packet comes through that requires the MAC entry.	command, then the command executed successfully; if the returned value is equal to any value other than the exact length of the issued command, then the command did not execute successfully.
[RFRSHARP]	"refresh arp table"	attempts to refresh all of the entries in the ARP table.	if the returned value is equal to the exact length of the issued command, then the command executed successfully; if the returned value is equal to any value other than the exact length of the issued command, then the command did not execute successfully.
[SETAUTH]	"set auth"	this is an old command that is no longer used in LVF version 7.1.189.	[This function cannot be used in the FIPS mode of operation.]
[LDTYPE]	"set ldtype"	sets load type so that when a switchover occurs, the brick knows what to do.	if the returned value is equal to the exact length of the issued command, then the command executed successfully; if the returned value is

Writing Commands…			
LSMS Function	**Service Call**	**Description**	**Service Output**
			equal to any value other than the exact length of the issued command, then the command did not execute successfully.
[WLSTATE]	"write load state"	sets the load state for use by the "read load state". (This is an old command that is no longer used in LVF version 7.1.189)	[This function cannot be used in the FIPS mode of operation.]
[BOOTFREEZE]	"zb"	prevent the brick from rebooting in the event that a fatal error occurs (aka a "panic"). This allows critical information to be retained on the screen long enough to read it.	if the returned value is equal to the exact length of the issued command, then the command executed successfully; if the returned value is equal to any value other than the exact length of the issued command, then the command did not execute successfully.
[REBOOT]	"zr"	force the brick to reboot.	if the returned value is equal to the exact length of the issued command, then the command executed successfully; if the returned value is equal to any value other than the exact length of the issued command,

Writing Commands…			
LSMS Function	**Service Call**	**Description**	**Service Output**
			then the command did not execute successfully.
[REDIRECT]	"redirect"	this is an old command that is no longer used in LVF version 7.1.189.	[This function cannot be used in the FIPS mode of operation.]
[AIPSEC]	"add ipsec"	add a pending Security Association to the specified zone. (prerequisite begin load).	if the returned value is equal to the exact length of the issued command, then the command executed successfully; if the returned value is equal to any value other than the exact length of the issued command, then the command did not execute successfully.
[ADIPSEC]	"add dynamic ipsec"	add an active Security Association to the specified zone.	if the returned value is equal to the exact length of the issued command, then the command executed successfully; if the returned value is equal to any value other than the exact length of the issued command, then the command did not execute successfully.
[DDIPSEC]	"delete	delete an active Security	if the returned

Writing Commands…			
LSMS Function	**Service Call**	**Description**	**Service Output**
	dynamic ipsec"	Association to the specified zone.	value is equal to the exact length of the issued command, then the command executed successfully; if the returned value is equal to any value other than the exact length of the issued command, then the command did not execute successfully.
[TRCTRACE]	"trace"	prints general debug trace help (disabled in production).	if the returned value is equal to the exact length of the issued command, then the command executed successfully; if the returned value is equal to any value other than the exact length of the issued command, then the command did not execute successfully.
[TRCDUMP]	"trace dump"	Prints a specific table (disabled in production).	if the returned value is equal to the exact length of the issued command, then the command executed successfully; if the returned value is equal to any value

Writing Commands…			
LSMS Function	**Service Call**	**Description**	**Service Output**
			other than the exact length of the issued command, then the command did not execute successfully.
[TRCLEVEL]	"trace level"	sets trace levels (disabled in production).	if the returned value is equal to the exact length of the issued command, then the command executed successfully; if the returned value is equal to any value other than the exact length of the issued command, then the command did not execute successfully.
[TRCENABLE]	"trace enable"	enables specific tracing (disabled in production).	if the returned value is equal to the exact length of the issued command, then the command executed successfully; if the returned value is equal to any value other than the exact length of the issued command, then the command did not execute successfully.
[TRCHELP]	"trace help"	prints general or specific debug trace help.	Displays control status information

Writing Commands…			
LSMS Function	**Service Call**	**Description**	**Service Output**
			about how to use the trace functions
[DUMPENABLE]	"dump enable"	causes a stack dump to be generated if the current thread terminates.	if the returned value is equal to the exact length of the issued command, then the command executed successfully; if the returned value is equal to any value other than the exact length of the issued command, then the command did not execute successfully.
[ARPSRVRS]	"arp servers"	causes the brick to generate ARPs for any local management addresses (i.e. LSMS).	if the returned value is equal to the exact length of the issued command, then the command executed successfully; if the returned value is equal to any value other than the exact length of the issued command, then the command did not execute successfully.
[ADDAUDFIL]	"add audit filter"	create an audit msg trace filter.	if the returned value is equal to the exact length of the issued command, then the command executed

Writing Commands…			
LSMS Function	**Service Call**	**Description**	**Service Output**
			successfully; if the returned value is equal to any value other than the exact length of the issued command, then the command did not execute successfully.
[MODAUDFIL]	"mod audit filter"	modify an audit msg trace filter.	if the returned value is equal to the exact length of the issued command, then the command executed successfully; if the returned value is equal to any value other than the exact length of the issued command, then the command did not execute successfully.
[DELAUDFIL]	"delete audit filter"	delete an audit msg trace filter.	if the returned value is equal to the exact length of the issued command, then the command executed successfully; if the returned value is equal to any value other than the exact length of the issued command, then the command did not execute successfully.

Writing Commands…			
LSMS Function	**Service Call**	**Description**	**Service Output**
[SETAUDFIL]	"set audit filter"	enable/disable an audit msg trace filter.	Enables or disables an audit msg trace filter. If the returned value is equal to the exact length of the issued command, then the command executed successfully; if the returned value is equal to any value other than the exact length of the issued command, then the command did not execute successfully.
[SETARPFILTER]	"set arp filter"	enable/disable arp filters.	Enable/disable ARP filters. If the returned value is equal to the exact length of the issued command, then the command executed successfully; if the returned value is equal to any value other than the exact length of the issued command, then the command did not execute successfully.
[SETNONIPFILTER]	set nonip filter"	enable/disable non-IP filters.	Enable/Disable non-IP filters.

Writing Commands…			
LSMS Function	**Service Call**	**Description**	**Service Output**
			If the returned value is equal to the exact length of the issued command, then the command executed successfully; if the returned value is equal to any value other than the exact length of the issued command, then the command did not execute successfully.
[ADDPKTFIL]	"add packet filter"	create a packet trace filter	If the returned value is equal to the exact length of the issued command, then the command executed successfully; if the returned value is equal to any value other than the exact length of the issued command, then the command did not execute successfully.
[MODPKTFIL]	"mod packet filter"	modifies a packet trace filter	If the returned value is equal to the exact length of the issued command, then the command executed successfully; if the

Writing Commands…			
LSMS Function	**Service Call**	**Description**	**Service Output**
			returned value is equal to any value other than the exact length of the issued command, then the command did not execute successfully.
[DELPKTFIL]	"delete packet filter"	deletes a packet trace filter	If the returned value is equal to the exact length of the issued command, then the command executed successfully; if the returned value is equal to any value other than the exact length of the issued command, then the command did not execute successfully.
[SETPKTFIL]	"set packet filter"	enables/disables a packet trace filter	If the returned value is equal to the exact length of the issued command, then the command executed successfully; if the returned value is equal to any value other than the exact length of the issued command, then the command did not execute successfully.

Writing Commands…			
LSMS Function	**Service Call**	**Description**	**Service Output**
[SETTHROTTLE]	"set throttle"	sets the size of the window over which error messages get throttled. (“throttled” means to have the message rate reduced to a particular level.)	If the returned value is equal to the exact length of the issued command, then the command executed successfully; if the returned value is equal to any value other than the exact length of the issued command, then the command did not execute successfully.
[WWHATAREU]	"what are you"	causes the brick to identify itself	Displays status information about the brick on screen
[DSESS]	"delete session"	deletes an entry from the session cache.	if the returned value is equal to the exact length of the issued command, then the command executed successfully; if the returned value is equal to any value other than the exact length of the issued command, then the command did not execute successfully.
[CONFIG]	"config"	implements a number of subcommands to modify or display: - Intelligent Cache Management Policy. - MAC move and starcast	Displays configuration information for description of subcommands.

Writing Commands…			
LSMS Function	**Service Call**	**Description**	**Service Output**
		zone matching policy. - UDP encapsulation policy - redundant LSMS rehome policy - SLA probes - the current (write) command tracing setting - also allows for removal of cache entries based upon the tag that associates them with a particular dynamic host group or IPSec tunnel.	If a subcommand is issued, then if the returned value is equal to the exact length of the issued command, then the command executed successfully; if the returned value is equal to any value other than the exact length of the issued command, then the command did not execute successfully.
[SMINOSCFG]	"switch minos"	move a couple of brick-wide configuration settings from pending to active (starcast zone matching & mac moves).	if the returned value is equal to the exact length of the issued command, then the command executed successfully; if the returned value is equal to any value other than the exact length of the issued command, then the command did not execute successfully.
[WFAILOVER]	"write failover"	display failover info or cause failover to standby.	if the returned value is equal to the exact length of the issued command, then the command executed successfully; if the

Writing Commands…			
LSMS Function	**Service Call**	**Description**	**Service Output**
			returned value is equal to any value other than the exact length of the issued command, then the command did not execute successfully. Displays status output failover information.
[CANFAILOVER]	"can failover"	examines the state of the standby to determine if it can take over all of the processing without losing anything (i.e. no interfaces have failed).	if the returned value is equal to the exact length of the issued command, then the command executed successfully; if the returned value is equal to any value other than the exact length of the issued command, then the command did not execute successfully.
[SETSFD]	"set file descriptor”	set the file descriptor associated with an active remote console.	if the returned value is equal to the exact length of the issued command, then the command executed successfully; if the returned value is equal to any value other than the exact length of the issued command, then the command

Writing Commands…			
LSMS Function	**Service Call**	**Description**	**Service Output**
			did not execute successfully.
[SETTRACEFLAG]	“set trace flag”	the flag controls whether or not certain messages (such as those generated using the trace audit command) get displayed on the console.	if the returned value is equal to the exact length of the issued command, then the command executed successfully; if the returned value is equal to any value other than the exact length of the issued command, then the command did not execute successfully.
[EFILEDOWN]	"exit fdownload”	force the thread that waits for the active brick to send it messages to quite so this brick can go active.	if the returned value is equal to the exact length of the issued command, then the command executed successfully; if the returned value is equal to any value other than the exact length of the issued command, then the command did not execute successfully.
[PORTTBL]	"add interface"	add interface information to the pending table (prerequisite begin tableload).	if the returned value is equal to the exact length of the issued command, then the command

Writing Commands…			
LSMS Function	**Service Call**	**Description**	**Service Output**
			executed successfully; if the returned value is equal to any value other than the exact length of the issued command, then the command did not execute successfully.
[VIPTBL]	"add vlanip"	add information about a VLAN (prerequisite begin tableload).	if the returned value is equal to the exact length of the issued command, then the command executed successfully; if the returned value is equal to any value other than the exact length of the issued command, then the command did not execute successfully.
[PARTITION]	"add partition"	adds a brick partition to the pending table. (prerequisite begin tableload).	if the returned value is equal to the exact length of the issued command, then the command executed successfully; if the returned value is equal to any value other than the exact length of the issued command, then the command did not execute

Writing Commands…			
LSMS Function	**Service Call**	**Description**	**Service Output**
			successfully.
[SETTIMEOFFSET]	"set timeoffset"	sets the time offset between the LSMS and the brick.	if the returned value is equal to the exact length of the issued command, then the command executed successfully; if the returned value is equal to any value other than the exact length of the issued command, then the command did not execute successfully.
[WTTCMDS]	"ctrl commands"	a collection of commands that display information about the amount of memory free, number of packets processed, etc.	tts - display the stack of the currently executing thread ttS - display the stacks of all of the threads. ttx - display a summary of memory usage ttd - exists in the API, but does nothing. ttp - displays per thread statistics and current state ttD - redisplays the last panic dump since the brick rebooted (if any) ttr - reboot the brick ttm - another memory usage summary ttq - display the mac

Writing Commands…			
LSMS Function	**Service Call**	**Description**	**Service Output**
			table tta - enable copying audit messages to the console as well as the LSMS ttb - toggle the "enable fastpkt" flag (fastpkt is a fast packet processing algorithm for TCP and UDP) ttE and ttP - make the brick print out usage statistics every 30 seconds. ttc - displays session cache statistics tt? - tt command help ttF - display syn flood table ttf - display list of files attached to thread #6.
[WBOOTDELAY]	"set bootdelay"	change the default internal delay from the time the brick boots until the time it can become active.	if the returned value is equal to the exact length of the issued command, then the command executed successfully; if the returned value is equal to any value other than the exact length of the issued command, then the command did not execute successfully.
[WADDAPPFILTER]	"add appfilter"	add an entry to the pending application filter policy	if the returned value is equal to

Writing Commands…			
LSMS Function	**Service Call**	**Description**	**Service Output**
		(prerequisite begin load).	the exact length of the issued command, then the command executed successfully; if the returned value is equal to any value other than the exact length of the issued command, then the command did not execute successfully.
[WPING]	"ping"	sends out pings.	Display status output ping information
[WTRACEROUTE]	"traceroute"	does traceroute.	Display status output traceroute information
[ADDAGGREGATE]	"add aggregate"	adds link aggregation information to the pending brick config table (prerequisite (begin tableload).	if the returned value is equal to the exact length of the issued command, then the command executed successfully; if the returned value is equal to any value other than the exact length of the issued command, then the command did not execute successfully.
[ADDPPPOE]	"add pppoe"	adds Point to Point Protocol over Ethernet (PPPoE) information to the pending brick config table. (prerequisite (begin	if the returned value is equal to the exact length of the issued command, then the

Writing Commands…			
LSMS Function	**Service Call**	**Description**	**Service Output**
		tableload).	command executed successfully; if the returned value is equal to any value other than the exact length of the issued command, then the command did not execute successfully.
[DISPLAYPPPOE]	"display pppoe"	displays current PPPoE state.	Displays current PPPoE state
[TRACEPPPOE]	"trace pppoe”	enables the brick to print PPPoE negotiation messages.	if the returned value is equal to the exact length of the issued command, then the command executed successfully; if the returned value is equal to any value other than the exact length of the issued command, then the command did not execute successfully.
[DISPLAYNONIP]	"display nonip"	displays the current non-IP protocols to allow through the brick.	Displays current non-IP protocols allowed with module
[INSTALLAGGREGATES]	"instaggr"	activates the currently pending link aggregation set without deleting the pending set.	if the returned value is equal to the exact length of the issued command, then the command executed successfully; if the

Writing Commands…			
LSMS Function	**Service Call**	**Description**	**Service Output**
			returned value is equal to any value other than the exact length of the issued command, then the command did not execute successfully.

Table 8 - LSMS Writing Commands

Reading Commands…			
LSMS Function	**Service Call**	**Description**	**Service Output**
[RRULES]	"read rules"	read the rules for a particular zone.	Displays rules for a particular zone.
[RTABLE]	"read table"	read the zone assignment table entries.	Displays table entries for zone assignment.
[RCACHE]	"read cache"	read the session cache entries or some summary info for a zone.	Displays session cache entries/summary information for a zone.
[RCONFIG]	"read config data"	read information about the defined management servers.	Display configuration information about defined management servers.
[RKEYWRD]	"read keyword"	read keywords from the inferno.ini configuration file.	Displays keywords from inferno.ini configuration file
[RTIME]	"read time"	the current timestamp.	Displays the current timestamp.
[RUPTIME]	"read uptime"	read the number of seconds since the brick booted/became active.	Displays the number of seconds since the module booted/became active.

Reading Commands…			
LSMS Function	**Service Call**	**Description**	**Service Output**
[REPORTICM]	"report icm"	read information about the state of the Intelligent Cache Management feature.	Displays status information about the state of the Intelligent Cache Management.
[RDOMINF]	"read dominfo"	read information about the policy's signer.	Displays information about policy's signer.
[RTBLINF]	"read tblinfo"	read information about the brick config's signer.	Displays information about module's configuration signer.
[RLSTATE]	"read load state"	this is an old command that is no longer used in LVF version 7.1.189	[This function cannot be used in the FIPS mode of operation.]
[RPINGSTAT]	"read ping stat"	read whether or not the audit channel seems healthy.	Displays ping status information.
[RSAS]	"read sas"	read some information about the SAs for a zone. (e.g. SPIs, host addresses, algorithms. *NOT* keys).	Displays SA information for a zone.
[REXPORT]	"get export"	read whether or not this brick is restricted to 56 bit encryption.	Displays status information on whether module is restricted to 56 bit encryption.
[RSWVERSION]	"get sw_version"	read the current software version.	Displays current software version.
[RMAC]	"read mac"	read entries from the MAC table.	Displays entries from MAC table.
[RARP]	"read arp"	read entries from the ARP table.	Displays entries from ARP table.
[RAUDFIL]	"read audit filter"	read entries from the audit trace filter table.	Displays entries from audit trace filter table.
[RPKTFIL]	"read packet filter"	read entries from the packet trace filter table.	Displays entries from the packet trace filter table.
[RHSTGRPS]	"read hostgroups"	read entries from the host group table for a zone.	Displays entries from the host group

Reading Commands…			
LSMS Function	**Service Call**	**Description**	**Service Output**
			table for a zone.
[RSRVGRPS]	"read servicegroups”	read entries from the service group table for a zone.	Displays entries from the service group table for a zone.
[RROUTES]	"read routes”	read the list of static routes.	Displays the list of static routes.
[MHASH]	"match hash"	determine whether the hash of a string matches a reference hash.	Displays whether the hash of a string matches a reference hash.
[RWHATAREU]	"what are you"	reads the brick's name and a couple of other useful pieces of information.	Displays module’s name, version, and other useful information about the module.
[RCOUNTDYNSAS]	"count dynamic sas"	displays the number of SA's loaded via the "add dynamic ipsec" command on this zone.	Displays number of SAs loaded via the “add dynamic ipsec” command on the zone.
[RMINOS]	"read minos"	displays information about the MAC move feature and the starcast zone matching policy.	Displays information on MAC move feature and the starcast zone matching policy.
[RACTIVITY]	"read activity"	reads information about whether the brick is ready to transition from standby to active.	Displays whether module is ready to transition from standby to active.
[RFAILOVER]	"read failover"	displays failover information.	Displays failover status.
[RDTHROTTLE]	"read throttle"	displays the current error message throttling interval.	Displays current error message throttling interval.
[RFILEDOWN]	"read fdownload"	waits for file transfer information from the active to the standby.	Displays file transfer information from active to standby.
[RSTTIMER]	"read stickiness timer" (LSMS redundancy)	reads how long the brick should wait before trying to go back to the higher	Displays how long the module should wait before trying

Reading Commands…			
LSMS Function	**Service Call**	**Description**	**Service Output**
		priority LSMS.	to get back to the higher priority LSMS.
[READ]	"read"	reads information about the current configuration for: - UDP encapsulation policy - NAT table policy - SLA probes	Displays current configuration information for: - UDP encapsulation policy - NAT table policy - SLA probes
[RVLANS]	"read vlans"	reads information about the VLAN configuration.	Displays VLAN configuration information.
[RPARTITIONS]	"read partitions"	reads information about the partition configuration.	Displays partition configuration information.
[RLASTHOMEDLSMS]	"read lastlsms"	reads what LSMS was last connected.	Displays what LSMS was last connected.
[RDEC64]	"read decode64"	reads the result of decoding base 64 encoded input back into its original form.	Displays result of decoding base 64 information.
[RENC64]	"read encode64"	reads the result of encoding base 64 arbitrary byte streams.	Displays result of encoding base 64 information.
[RCONTACT]	"read audit contact"	reads whether or not the audit channel is active.	Displays whether or not the audit channel is active.
[RRANDOM]	"get random bytes"	reads some pseudo random bytes. Used during the initialization of flash.	Sends back a pseudo random number to be used.
[DHCP]	"dhcp"	displays current DHCP client state.	Displays current DHCP client state.
[RMODELNUMBER]	"read model"	displays the model number of this brick.	Displays the module’s model number.
[VPN]	"vpn"	disabled on this version of the brick.	N/A

Table 9 - LSMS Reading Commands

The console/serial/keyboard/monitor ports provide a CLI which offers the Crypto Officer the following services:

Service Input	Description	Service Output
“bootstrap”	allows CO to reload the certificate and initialization information into the brick via the serial port (keyboard)	Bootstraps the module
“help”	prints list of commands	Displays list of commands and their system usage
“help <cmd>”	prints help for <cmd>	Displays usage of <cmd>
“logout”	logout from remote port	Closes down the CLI
“initialize flash”	initializes flash configuration	Initializes the flash configuration
“ping [options]”	sends an ICMP ping packet and prints response times	sends ICMP ping to specified IP address
“repeat”	repeat the previous command	Attempts to execute the previous command entered by keyboard
“refresh <table>” table	refresh brick’s mac or arp	Displays “<table> table cleared if successful” Displays “Error -> refresh, missing table <mac, arp> argument” if unsuccessful
“display arptable”	display contents of the arp table	Displays the IP Address, MAC Address, VlanID, Status, Refcntarptable, and total arp entries
“display configuration”	prints the inferno.ini file	Displays the contents of the inferno.ini file
“display dhcp”	display DHCP configuration information	Displays DHCP server IP, DHCP gateway IP, time lease expires in, time lease renewal in, and DNS server(s)
“display encapsulation <zone>”	display UDP encapsulation info for the zone	Displays the UDP encapsulation information for the <zone>
“display failover”	display failover status	Displays failover status if enabled;

Service Input	Description	Service Output
		Displays “Failover feature not enabled” if disabled
“display files <filepath>”	print the names of the files	Displays the size, date, and names of the files for the given <filepath>
“display hostgroups <zone>”	display a zone’s hostgroup definitions	Displays a table with Host Name, Typ, TmOut, TagValue, IP Address / Range for all entries in the <zone>
“display icm”	display ICM info	Displays current ICM information
“display interfacestatus [<if>]”	display information about an interface’s NIC	Displays the Interface, Root, I/F, MAC, Link, Speed, and Mode for all the interfaces on the NIC
“display lsms”	print the current LSMS connected (or the last LSMS)	Displays “Last LSMS was <last IP address of LSMS>”
“display mactable [<if>]”	display MAC table for the specified interface	Displays a table with entries for IF, MAC, Address, Status, VLAN, and Refcnt for all mac table entries and total number of mac table entries
“display mempools”	print information on 5 memory pools of the brick	Displays information on the memory pools of the brick in a table as Pool, Max-Size, Cur-Size, Peak, Arena-Sz, and In-Use
“display nat <zone>”	print information about NAT tables for a zone	Displays a table with entries for Name, RefCt, Pre-NAT list, and Post-NAT list
“display partitions”	print partition information	Displays partition and VLAN ID
“display policy < zone>”	prints the ruleset for the specified zone	Displays a table with entries for Rule#, Source, Destination, Service, A, D, SM, DM, PM, DEP, and VPN. Displays load date, sign date, and LSMS administrator for the

Service Input	Description	Service Output
		policy.
“display pppoe”	display pppoe information	Displays pppoe information for #, Vlan, States, Address, MTU, DNS1, and DNS2
“display remoteconsole”	display information about the remote console	Displays “User <user> is connected through remote console.”
“displayroutes [<if>]”	display routing information for an interface	Displays routing information for an interface
“display sa <zone>”	display a zone’s current security associations	Displays SPI, User Name, Source, Destination, Prot, AH, ESP, TEP, Sec/Kbytes for current SAs
“display servicegroups <zone>”	display a zone’s servicegroup definitions	Displays Service, Name, Definitions, and App Mon for <zone>
“display sessions <zone> [<IP-addr>]	prints the zone’s session cache optionally filtered by an IP address	Displays Source, Destination, Service, AVE, Rule#, FWD-PKT/B, and REV-PKT/B for <zone>
“display slamon <zone> ”	displays the list of SLA probes and some statistics about each one (#send, #received, max round trip delay)	Displays #send, #received, max round trip, delay for entries in <zone> if they exist
“display time”	print the brick’s current time in GMT	Displays “the current time is <date> <time> GMT” Displays “Active since <date> <time>.” Displays time from GMT and Brick local time
“display version”	print the bricks’ version number	Displays Softw vers, VPN cards (if any), Status, Starcast zone, and MAC moves
“display vlans”	display vlan ip subnets and port membership	Displays VLAN, IP, Address, Mask, and Ports for each VLAN
“display zonetable”	display the brick’s zone assignment table	Displays table with entries for Ifc, Address, Range, Zone, VLAN, and

Service Input	Description	Service Output
		VBA. Displays date and time, signer of policy, software version, VPN cards (if any), status, starcast zone and MAC moves
“failover”	Toggles failover	Failovers, if failover is enabled (from configuration)
“failover yield [force]”	switch from active to standby	Switches from active to standby
“traceroute”	trace the network route used by brick when sending	Displays hops, time, and bytes
“trace arp on [yes|no]”	trace arps (with optional full packet dump)	Displays “Tracing of arp enabled”
“trace arp off”	disable arp tracing	Displays “Tracing of arp disabled”
“trace audit filter <filter-list>”	define an audit filter	Defines an audit filter
“trace audit modify <filter-id> <filter-list>”	modify existing audit filter	Modifies existing audit filter
“trace audit delete <filter>”	delete the specified filter	Deletes specified filter
“trace audit on [<filter-id|a|p>]”	enable all filters or the specified filter	Displays “<filter-id\a\p> enabled.” and information for filers
“trace audit off [<filter-id|a|p>]”	disable all filters or the specified filter	Displays “<filter-id\a\p> disabled.”
“trace packet list”	print the list of current packet filters	Displays table with entries for #, E, D, IF, Source, Destination, Protocol, and Format
“reboot [<msg>]”	reboots the module with an optional message in the audit log	Reboots module. Reboots module with message, if <msg> is given
“set screensize [<size>]”	set or display the screensize, default=23	Sets screensize to <size>
“set printing [on|off]”	set or display the tracing print value	Sets or displays tracing print value
“set baudrate <rate>”	set or display the baudrate of the remote port	Sets or displays baudrate of remote port
“set throttle <interval>”	set number of seconds b/w identical audit msgs	Sets number of seconds between identical audit messages as given by <interval>

Service Input	Description	Service Output
“set errors [on|off]”	set or display the critical error value	Set or displays critical error values
“modem <cmd>”	send the <cmd> to the brick’s modem, use “ to enclose blanks	Sends <cmd> to the brick’s modem

Table 10 – CLI Service Commands

2.3.2 User Services

The User role has access to the IPSec services of the module as a VPN client. The module provides the User role to Remote Users. A User authenticates to the module per packet using the shared secret HMAC-SHA-1 key configured by the Crypto Officer. Through IPSec, the User role has access to some of the module’s cryptographic functionality, including 3DES encryption/decryption and HMAC-SHA-1 calculation/verification.

2.4 Physical Security

All models are contained within a strong steel production-grade enclosure with serialized tamper evident labels. Upon tampering, the tamper evident labels will clearly provide tamper evidence via tears, chips, or cracking. On-board LAN connectors, console connectors, power supply switches, and power cable connections are provided on the rear of the Brick 350 and Brick 1000. Power switches and floppy drives are provided on the front of all models.

Once the module has been configured to meet FIPS 140-2 Level 2 requirements, the module cannot be accessed without signs of tampering. To seal the module, apply serialized tamper-evidence labels as follows:

Brick 350 Module:

1. Clean the cover of any grease, dirt, or oil before applying the tamper evidence labels. Alcohol-based cleaning pads are recommended for this purpose. The temperature of the Module should be above 10°C.
2. Having the front cover facing you, place the first label on the top left of the chassis as shown in Figure 10. The tamper evidence label should be placed so that one half of the tamper evidence label covers the top of the chassis and the other half covers the left side of the chassis. Any attempt to remove the enclosure will leave tamper evidence.
3. Place the second label on the chassis as shown in Figure 10. The tamper evidence label should be placed so that one half of the tamper evidence label covers the top of the chassis and the other half covers the right side of the chassis. Any attempt to remove the enclosure will leave tamper evidence.

4. Place the third label over the floppy drive as shown in Figure 10. The tamper evidence label should be placed so that one half of the tamper evidence label covers the front top of the chassis over the floppy drive and the other half covers the floppy drive. Any attempt to remove the enclosure or insert or remove a floppy disk will leave tamper evidence.
5. The labels completely cure within five minutes.

Brick 1000 Module:

1. Clean the cover of any grease, dirt, or oil before applying the tamper evidence labels. Alcohol-based cleaning pads are recommended for this purpose. The temperature of the Module should be above 10°C.
2. Having the front cover facing you, place the first label on the top left of the module as shown in Figure 11. The tamper evidence label should be placed so that the one half of the tamper evidence label covers the top of the chassis and the other half covers the left side of the chassis. Any attempt to remove the enclosure will leave tamper evidence.
3. Place the second label on the chassis as shown in Figure 11. The tamper evidence label should be placed so that one half of the tamper evidence label covers the top of the chassis and the other half covers the right side of the chassis. Any attempt to remove the enclosure will leave tamper evidence.
4. Place the third label on the chassis as shown in Figure 11. The tamper evidence label should be placed so that the one half of the tamper evidence label covers the top of the front panel cover of the chassis and the other half covers the front of the top of the chassis. Any attempt to remove the enclosure or open the front panel cover will leave tamper evidence.
5. Place the fourth label on the back right of the module as shown in Figure 11. The tamper evidence label should be placed so that the one half of the tamper evidence label covers the back panel of the enclosure and the other half covers the first power supply. Any attempt to remove the enclosure will leave tamper evidence.
6. Place the fifth label on the back right of the module as shown in Figure 11. The tamper evidence label should be placed so that the one half of the tamper evidence label covers the back panel of the enclosure and the other half covers the redundant power supply. Any attempt to remove the enclosure will leave tamper evidence.
7. Place the sixth label on the back of the module as shown in Figure 11. The tamper evidence label should be placed so that the tamper evidence label covers the far left screw, the chassis, and the top of the opacity shield. Any attempt to remove the enclosure will leave tamper evidence.
8. Place the seventh label on the back of the module as shown in Figure 11. The tamper evidence label should be placed so that the tamper evidence label covers the middle screw, the chassis, and the top of the opacity shield. Any attempt to remove the enclosure will leave tamper evidence.
9. Place the eighth label on the back of the module as shown in Figure 11. The tamper evidence label should be placed so that the tamper evidence label covers the far right screw, the chassis, and the top of the opacity shield. Any attempt to remove the enclosure will leave tamper evidence.
10. The labels completely cure within five minutes.

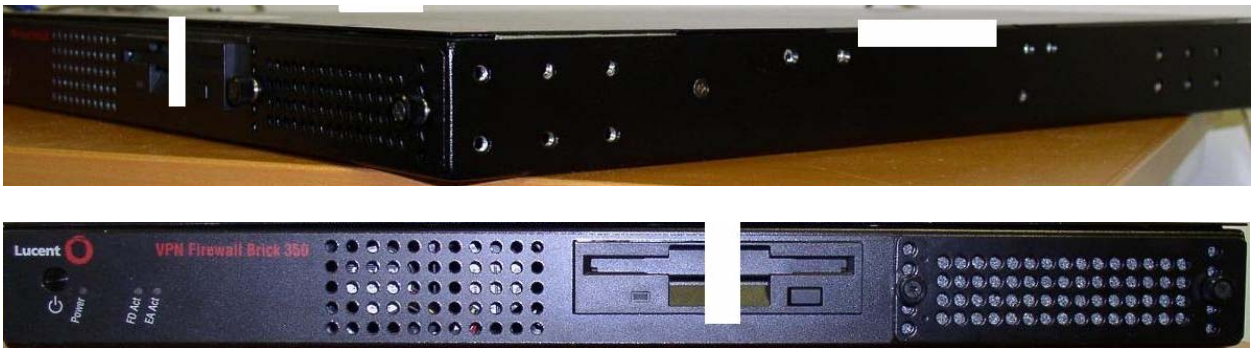

Figure 10 – Brick 350 Tamper Evidence Label Placement

Figure 11 – Brick 1000 Tamper Evidence Label Placement

The tamper evidence seals are produced from a special thin gauge vinyl with self-adhesive backing. Any attempt to open the module or remove the front faceplate will damage the tamper evidence seals or the painted surface and metal of the module cover. Since the tamper evidence seals have non-repeating serial numbers, they may be inspected for damage and compared against the applied serial numbers to verify that the module has not been tampered. Tamper evidence seals can also be inspected for signs of tampering, which include the following: curled corners, bubbling, crinkling, rips, tears, and slices. The word "OPEN" may appear if the label was peeled back.

2.5 Cryptographic Key Management

The module securely administers both cryptographic keys and other critical security parameters (CSPs) such as passwords. The tamper evidence seals provide physical protection for all keys. Keys are also password protected and can be zeroized by the Crypto Officer. Keys are exchanged manually only.

The following table shows the module's critical security parameters and the type of access that each role has to them:

SRDI/Role/Service Access Policy	Security Relevant Data Item	CSP 1 - IPSec Session Keys	CSP 2 – Crypto Officer Session Key	CSP 3 – Crypto Officer CA public key	CSP 4 – DSA key pair for Module	CSP 5 – Diffie-Hellman key pairs	CSP 6 – PRNG Seed Key	CSP 7 – Crypto Officer Password Hash
Role/Service								
User Role								
IPSec Functions		r						
Crypto-Officer Role								
Initialize Module		w	w	w	w	w		w
Manage Module		r w d	r w d	r w d	r w d	r w d	d	r w d

Table 11 - Module's CSPs and Access Types

The module supports DES, 3DES, SHA-1, HMAC-SHA-1, MD5, HMAC-MD5, ARC4 (functionally equivalent to RC4), Diffie-Hellman, and DSA (for digital signatures) cryptographic

algorithms. The MD5, HMAC-MD5, and ARC4 encryption/decryption algorithms are disabled when operating in FIPS mode.

The Module has the following CSPs:

1) IPSec Session Keys - VPN Tunnel keys (DES or 3DES – NIST FIPS PUB 46-3 and HMAC – NIST PUB 198)

2) Crypto Officer Session keys - Session keys for Crypto Officer (DES or 3DES – NIST FIPS PUB 46-3 and HMAC – NIST PUB 198)

3) FIPS Approved PRNG seed key (FIPS 186-2 Appendix 3.2 and 3.3 with Change Notice 1) which is used in the generation of all keys. (The FIPS Approved PRNG is used for all cryptographic and security relevant operations. The non-Approved non-deterministic RNG is used to seed the FIPS approved PRNG.)

4) Crypto Officer CA DSA public key (NIST FIPS PUB 186-2 in a certificate)

5) DSA Key Pair for Module (NIST FIPS PUB 186-2 with public key in a certificate)

6) Diffie-Hellman key pairs

7) A Key for talking to the LSMS for Firewall User Authentication or to the Lucent Proxy Agent (LPA) for reflection (DES – NIST FIPS PUB 46-3)

8) A Key for proxied Internet Key Exchange (IKE) messages – to ensure that messages from the LSMS to the module are not spoofed. (SHA-1 hash of the Module's certificate – PUB 180-1)

Critical Security Parameters (CSPs) Zeroization:

IPSec Session Keys (VPN Tunnel keys) come in two flavors: manual and automatic. Manual keys are defined by the administrator and downloaded as part of the policy (and hence stored in the flash disk as well as in memory). When the tunnel is deleted by the administrator, the memory specifically allocated to the tunnel keys is zeroed. When the box is powered down, all copies of the keys in all memory is destroyed. Destruction of the key in flash requires wiping the flash disk. Automatic keys do not have a copy on the flash, but are otherwise identical.

The flash disk may be zeroized by inserting a floppy containing the FIPS-approved LVF image and booting the module. The Crypto Officer must follow the onscreen instructions to install a new boot sector and zeroize the flash.

Crypto Officer Session keys only exist in memory (never in flash). They are zeroized when the session is lost or when the box is powered down. The Crypto Officer Password Hash is zeroized when the flash is zeroized.

The key for talking to the LSMS for Firewall User Authentication or the LPA for reflection is configured by the administrator as part of the policy and hence stored in the flash disk as well as in memory. Zeroization of this key is identical to the manual IPSec Session key.

The key for proxied IKE messages is used only as address spoof protection of the proxied IKE messages between the module and the LSMS. This key can be zeroized by wiping the flash disk.

2.6 Self-Tests

In order to prevent any secure data from being released, it is important to test the cryptographic components of a security module to ensure that all components are functioning correctly. The module includes an array of self-tests that are run during startup and periodically during operations.

The module's Power-up and Conditional Tests (software and hardware) consist of the following:

- DES Known Answer Test
- 3DES Known Answer Test
- SHA-1 Known Answer Test
- HMAC-SHA-1 Test
- Software/Firmware Load Test
- Software/Firmware Integrity Test
- Power-up Self-test for DSA
- Continuous Random Number Generator Test
    - FIPS approved CRNG
    - Non-FIPS approved CRNG (used as the seed key to the PRNG)
- Critical Functions Test
    - Diffie-Hellman Known Answer Test
    - RNG Known Answer Test
    - Bypass Test

If any one of the self-tests fail, the module transitions into an error state. Within the error state, all secure data transmission is halted and the module outputs status information indicating the failure.

3 Secure Operation of the Brick 350 and Brick 1000 VPN Firewalls

The Brick 350 and Brick 1000 modules meet all the Level 2 requirements for FIPS 140-2. Follow the setting instructions provided below to place the module in FIPS mode. Operating this module without maintaining the following settings will remove the module from the FIPS approved mode of operation.

3.1 Initial Setup

1. After removing the module from the packaging, the Crypto Officer must verify that the tamper evidence warranty sticker(s) have not been compromised. If the warranty sticker(s) shows signs of tampering, then the Crypto Officer shall consider the module to have been compromised in transit and must not use the module. The Crypto Officer shall contact Lucent for further instructions.
2. The Crypto Officer must apply tamper evidence labels as described in Section 2.4 of this document.
3. Only a Crypto Officer may open the chassis. When removing the tamper evidence label, the Crypto Officer should remove the entire label from the module and clean the cover of any grease, dirt, or oil with an alcohol-based cleaning pad. The Crypto Officer must re-apply tamper evidence labels on the module as described in Section 2.4.
4. For the Brick 1000 module: The Crypto Officer must apply the opacity shield as described in Section 2.1 of this document.

3.2 Module Initialization and Configuration

1. The Crypto Officer must perform the initial configuration. Lucent LVF version 7.1.189 is the only allowable image; no other image may be loaded.
2. The Crypto Officer must include the following command in the configuration file (inferno.ini):

   ```
   fips=y
   ```

   (Note: The Crypto Officer may use the Lucent LSMS for this purpose. The Crypto Officer may toggle the FIPS checkbox from the ‘Brick Editor’ screen under the ‘Options’ tab.)
3. The Crypto Officer must set a password on the Console/Serial/Keyboard/Monitor ports via the following configuration file (“inferno.ini”) command:

   RemoteLoginId=<*SHA-1 hash of the desired password*>

   (Note: The Crypto Officer may use the Lucent LSMS for this purpose. The Crypto Officer may toggle the 'Enable Serial Port' checkbox and type in a password, followed by a verification of the same password.)
4. The Crypto Officer must not execute the “bootstrap”, “adproxy”, “setauth”, “wlstate”, “redirect”, and “rlstate” commands while the module is in a FIPS approved mode of operation.
5. For detailed instructions on the installation and configuration process, please see Chapter 3 of the Lucent Security Management Server, Administration Guide.

3.3 IPSec Requirements and Cryptographic Algorithms

1. There is one type of key management method that is allowed in FIPS mode: IPSec manually entered keys.
2. The following algorithms are not FIPS approved and must be disabled:
    - ARC4 for encryption
    - MD-5 for hashing and HMAC MD-5
3. The Crypto Officer must not use the Failover configuration in the FIPS approved mode of operation.

3.4 Remote Access

1. Crypto-Officers authenticate to the module during an SSL-like protocol. The Crypto Officer has a digital certificate associated with it. This certificate is used during the SSL-like handshake to authenticate the Crypto Officer to the module.

   The Module SSL-like authentication consists of several messages. These messages are used to negotiate connection parameters between the Crypto Officer and the module, exchange Crypto Officer and module certificates and public keys, and connection security parameters and authentication data.

   The Module SSL-like authentication consists of these 6 messages:
   Message 1: protocol version (always '2')
   Message 2: alpha**r0 mod p
   Message 3: sender's certificate (public portion)
   Message 4: Sender's public key.
   Message 5: alpha**r0 and alpha**r1 (far end's alpha**r0) signed with local secret key (which recipient verifies using public key)
   Message 6: acknowledgement + desired key strength + digest of same (using new shared secret key)

   The sequence of messages occurs between the Module and the LSMS as follows:

<<Module>>	<<LSMS>>
Message 1 →	
	← Message 1
Message 2,3,4 →	
	← Message 2,3,4
Message 5 →	
	← Message 5
Message 6 →	
	← Message 6

   After this exchange, initial authentication is complete.

2. Users authenticate to the module using a shared secret HMAC-SHA-1 key. This authentication is per packet via verification of an HMAC.